This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS

1. – 19. (canceled)

20.   (currently amended) A method of ~~distributed processing~~ storage and processing by a processor, comprising:

providing a memory having a plurality of memory segments capable of storing either program code or data;

providing a storage location for capable of storing semaphore values each of which are associated with one of said memory segments and operable to indicate whether said associated memory segment contains program code or data that is available for use by the processor;

providing a first program and a second program each operable to access the semaphore values;

accessing a first semaphore value by said first program;

determining if the program code or data in the memory segment associated with the first semaphore value is available for use by the processor based upon the first ~~semifore~~ semaphore value; and

using the first program to implement the code or data stored in the memory segment associated with the first semaphore value by the processor;

wherein the program code or data stored in the memory segment associated with the first semaphore value represents one of a plurality of blocks of program code or data collectively representing an algorithm too large to be stored in the memory segment associated with the first semaphore value, thereby enabling execution of the algorithm using limited amount of memory.

21.   (previously presented) A method as recited in claim 20, comprising:

altering the first semaphore value by the first program so as to indicate that the memory segment associated with the first semaphore value is available for having program code or data stored therein.

22.   (previously presented) A method as recited in claim 21, comprising:

accessing the altered first semaphore value by the second program;

determining if the memory segment associated with the altered first semaphore value is available to have program code or data stored therein by the second program; and

using the second program to store code or data in the memory segment associated with the semaphore value when the associated memory segment is available.

23. (previously presented) A method as recited in claim 22, comprising:

completing the storing of the program code or data in the memory segment associated with the semaphore value by the second program; and

altering the semaphore value by the second program to indicate that the program code or data in the memory segment associated with the first semaphore value is available for use by the first program.

24. (previously presented) A method as recited in claim 20, wherein the memory is a local memory of the processor and wherein the memory segments are logical memory segments implemented by the processor based upon a computer program to be executed by the processor.

25. (previously presented) A method as recited in claim 20 wherein the storage location is a register, or portions thereof, of the processor or a scalar accessible to the processor.

26. (previously presented) A method as recited in claim 20, wherein the first program is a routine that is located in a reserved portion of the local memory such that the first program can not be written over with previously presented code or data, and wherein the first program is operable to access the code or data stored in the local memory of the processor as well as implement that accessed code or data.

27. (previously presented) A method as recited in claim 26, wherein when the program that is operable to implement a portion of a Fast Fourier Transform (FFT) program is stored in the local memory, the first program is operable to access the local memory and begin implementing the FFT code.

28. (previously presented) A method as recited in claim 27, wherein the second program is operable to load previously presented blocks of code or data that are used by the first program.

29. (previously presented) A method as recited in claim 28, wherein the second program loads code or data from an external memory to the local memory based upon the first semaphore value.

30. (previously presented) A method as recited in claim 20, wherein the determining further comprises:

comparing the first semaphore value with a lookup list of pre-determined semaphore values.

31. (currently amended) Computer program product for ~~distributed processing~~ storage and processing by a processor, comprising:

computer code for storing either program code or data in a memory having a plurality of memory segments;

computer code for providing a storage location capable of storing semaphore values each of which are associated with one of said memory segments and operable to indicate whether said associated memory segment contains program code or data that is available for use by the processor;

computer code for providing a first program and a second program each operable to access the semaphore values;

computer code for accessing a first semaphore value by said first program;

computer code for determining if the program code or data in the memory segment associated with the first semaphore value is available for use by the processor based upon the first ~~semifore~~ semaphore value;

computer code for using the first program to implement the code or data stored in the memory segment associated with the first semaphore value by the processor; and

computer readable medium for storing the computer code;

wherein the program code or data stored in the memory segment associated with the first semaphore value represents one of a plurality of blocks of program code or data collectively representing an algorithm too large to be stored in the memory segment associated with the first semaphore value, thereby enabling execution of the algorithm using limited amount of memory.

32. (previously presented) Computer program product as recited in claim 31, comprising:

Atty. Dkt. No.: GENSP035   Page 4 of 12   Application No.: 09/672,517

PAGE 6/14 * RCVD AT 10/18/2004 6:31:44 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/3 * DNIS:8729306 * CSID:16509618301 * DURATION (mm-ss):04-06

computer code for altering the first semaphore value by the first program so as to indicate that the memory segment associated with the first semaphore value is available for having program code or data stored therein.

33. (previously presented) Computer program product as recited in claim 32, comprising:
computer code for accessing the altered first semaphore value by the second program;
computer code for determining if the memory segment associated with the altered first semaphore value is available to have program code or data stored therein by the second program; and
computer code for using the second program to store code or data in the memory segment associated with the semaphore value when the associated memory segment is available.

34. (previously presented) Computer program product as recited in claim 33, comprising:
computer code for completing the storing of the program code or data in the memory segment associated with the semaphore value by the second program; and
computer code for altering the semaphore value by the second program to indicate that the program code or data in the memory segment associated with the first semaphore value is available for use by the first program.

35. (previously presented) A method as recited in claim 31, wherein the memory is a local memory of the processor and wherein the memory segments are logical memory segments implemented by the processor based upon a computer program to be executed by the processor.

36. (previously presented) Computer program product as recited in claim 31 wherein the storage location is a register, or portions thereof, of the processor or a scalar accessible to the processor.

37. (previously presented) Computer program product as recited in claim 31, wherein the first program is a routine that is located in a reserved portion of the local memory such that the first program can not be written over with previously presented code or data, and wherein the first program is operable to access the code or data stored in the local memory of the processor as well as implement that accessed code or data.

38. (previously presented) Computer program product as recited in claim 37, wherein when the program that is operable to implement a portion of a Fast Fourier Transform (FFT) program is stored in the local memory, the first program is operable to access the local memory and begin implementing the FFT code.

39. (previously presented) Computer program product as recited in claim 38, wherein the second program is operable to load previously presented blocks of code or data that are used by the first program.

40. (previously presented) Computer program product as recited in claim 39, wherein the second program loads code or data from an external memory to the local memory based upon the first semaphore value.

41. (previously presented) Computer program product as recited in claim 31, wherein the determining further comprises:

computer code for comparing the first semaphore value with a lookup list of pre-determined semaphore values.

42. (currently amended) In a single processor system that includes a memory coupled to the processor having a plurality of memory segments capable of storing either program code or data, a method of ~~distributed processing~~ storage and processing by the processor, comprising:

storing semaphore values in the memory each of which are associated with one of said memory segments and operable to indicate whether said associated memory segment contains program code or data that is available for use by the processor;

accessing a first semaphore value by a first program;

based upon the first semaphore value, determining if the program code or data in the memory segment associated with the first semaphore value is available for use by the processor; and

implementing the code or data stored in the memory segment associated with the first semaphore value by the processor using the first program;

wherein the program code or data stored in the memory segment associated with the first semaphore value represents one of a plurality of blocks of program code or data collectively representing an algorithm too large to be stored in the memory segment

associated with the first semaphore value, thereby enabling execution of the algorithm using limited amount of memory.

43.   (previously presented)A method as recited in claim 42, comprising:

altering the first semaphore value by the first program so as to indicate that the memory segment associated with the first semaphore value is available for having program code or data stored therein.

44.   (previously presented)A method as recited in claim 43, comprising:

storing code or data in the memory segment associated with the first semaphore value with the altered semaphore value indicates that the memory segment is available by a second program.

45.   (previously presented)A method as recited in claim 44, comprising:

altering the semaphore value by the second program to indicate that the program code or data stored in the memory segment associated with the first semaphore value is available for use by the first program.

46.   (previously presented)A method as recited in claim 42, wherein the memory is a local memory of the processor and wherein the memory segments are logical memory segments implemented by the processor based upon a computer program to be executed.

47.   (currently amended)   In a single processor system that includes a memory coupled to the processor having a plurality of memory segments capable of storing either program code or data, computer program product for ~~distributed processing~~ storage and processing by the processor, comprising:

computer code for storing semaphore values in the memory each of which are associated with one of said memory segments and operable to indicate whether said associated memory segment contains program code or data that is available for use by the processor;

computer code for accessing a first semaphore value by a first program;

computer code for determining if the program code or data in the memory segment associated with the first semaphore value is available for use by the processor based upon the first semaphore value;

computer code for implementing the code or data stored in the memory segment
associated with the first semaphore value by the processor using the first program; and

computer readable medium for storing the computer code;

wherein the program code or data stored in the memory segment associated with the
first semaphore value represents one of a plurality of blocks of program code or data
collectively representing an algorithm too large to be stored in the memory segment
associated with the first semaphore value, thereby enabling execution of the algorithm
using limited amount of memory.

48.   (previously presented)Computer program product as recited in claim 47, comprising:

altering the first semaphore value by the first program so as to indicate that the memory
segment associated with the first semaphore value is available for having program code or data
stored therein.

49.   (previously presented)Computer program product as recited in claim 48, comprising:

computer code for storing code or data in the memory segment associated with the first
semaphore value with the altered semaphore value indicates that the memory segment is
available by a second program.

50,   (previously presented)Computer program product as recited in claim 49, comprising:

computer code for altering the semaphore value by the second program to indicate that
the program code or data stored in the memory segment associated with the first semaphore value
is available for use by the first program.

51.   (Currently amended) Computer program product as recited in claim 47, wherein the
memory is a local memory of the processor and wherein the memory segments are logical
memory segments implemented by the processor based upon a computer program to be executed.